Méthodes numériques : optimisation.
L3 2015–2016 — 2e semestre.
Feuille de TD n° 3 : Méthode du gradient conjugué.

1 Étude de fonction quadratique abstraite.

Soit E un espace vectoriel euclidien (c'est à dire de dimension finie et muni d'un produit scalaire noté $\langle \cdot, \cdot \rangle_E$). On se donne un endomorphisme $\mathcal{A}$ (linéaire de E dans E) et un élément b de E. On considère la fonction quadratique $E \to \mathbb{R}$ donnée par

$$f(x) = \frac{1}{2}\langle x, \mathcal{A}(x)\rangle_E + \langle b, x\rangle_E.$$

(a) Montrer que si on note A la matrice de $\mathcal{A}$ dans une base orthonormale de E, alors le fait que A soit symétrique est équivalent à :

$$\forall x, y \in E, \langle x, \mathcal{A}(y)\rangle_E = \langle y, \mathcal{A}(x)\rangle_E.$$

Montrer alors que dans ce cas, on a $\nabla f(x) = \mathcal{A}(x) + b$.

(b) On suppose que l'endomorphisme $\mathcal{A}$ vérifie la relation précédente et que pour tout $x \in E \setminus 0$, on a $\langle x, \mathcal{A}(x)\rangle_E > 0$. Montrer que si on pose $r = \mathcal{A}(x) + b$ et que $r \neq 0$, alors la fonction $t \mapsto f(x - t\,r)$ est un polynôme de degré deux à coefficient dominant strictement positif, et calculer son unique minimum.

(c) En déduire une expression des itérées de l'algorithme de descente de gradient à pas optimal pour f, qui permette d'utiliser seulement une évaluation de $\mathcal{A}$ et deux calculs de produits scalaires par étape (on notera $r_k = \nabla f(x_k)$).

2 Problème d'application : inpainting.

Le problème d'« inpainting » est le suivant : on se donne une image détériorée (provenant d'une image originale dont certains pixels n'ont pas été transmis), ainsi que la liste des pixels dont on est sûr qu'ils on bien été transmis correctement. On aimerait « interpoler » entre ces pixels connus de façon à remplir les zones indéterminées.

On modélise l'image par une matrice $F = (f_{i,j}) \in M_{n_1,n_2}(\mathbb{R})$, où $f_{i,j} \in [0, 1]$ est la valeur de la luminosité au pixel de la ligne i et colonne j, et n_1 (resp. n_2) est le nombre de lignes (resp. de colonnes) de pixels. Et on va noter $M = (m_{i,j})$ la matrice

du masque, c'est à dire $m_{i,j} \in \{0,1\}$ avec $m_{i,j}$ qui vaut 1 si le pixel a été transmis et 0 si le pixel n'a pas été transmis. Ce sont les données du problème.

Pour des raisons pratiques, on notera $\overline{M} = (\overline{m}_{i,j}) = (1 - m_{i,j})$ le contraire du masque, c'est-à-dire dont les coefficients valent 1 si le pixel n'a pas été transmis et 0 sinon.

Le but est de retrouver une image complète, qui corresponde le plus possible à l'image originale. Pour cela on cherche donc une matrice $U = (u_{i,j})$ qui soit telle que $u_{i,j} = f_{i,j}$ si le pixel a été transmis et qui soit la plus « lisse » possible, elle va minimiser une sorte de norme L^2 d'une discrétisation du gradient. On considère donc le problème d'optimisation suivant :

$$\inf_{U \in C} \frac{1}{2} \sum_{i=1}^{n_1-1} \sum_{j=0}^{n_2-1} (u_{i,j} - u_{i-1,j})^2 + \frac{1}{2} \sum_{i=0}^{n_1-1} \sum_{j=1}^{n_2-1} (u_{i,j} - u_{i,j-1})^2,$$

où l'ensemble des contraintes est

$$C = \{U \in M_{n_1,n_2}(\mathbb{R}) | \forall (i,j) \text{ tels que } m_{i,j} = 1, \text{ on a } u_{i,j} = f_{i,j}\}.$$

On a fait commencer les différents indices à 0, pour être cohérent avec Python.

(a) Montrer qu'en faisant le changement de variable $X = U - F \circledast M$ (où $\circledast$ est la multiplication élément par élément, c'est-à-dire que $(F \circledast M)_{i,j} = f_{i,j} m_{i,j}$), on obtient que $U \in C \Leftrightarrow X \in E$, où E est un sous-espace vectoriel de $M_{n_1,n_2}(\mathbb{R})$. Comment calculer la dimension n de E à partir de la matrice $\overline{M}$?

Pour $U, V \in M_{n_1,n_2}(\mathbb{R})$, on note

$$\varphi_2(U,V) = \sum_{i=1}^{n_1-1} \sum_{j=0}^{n_2-1} (u_{i,j} - u_{i-1,j})(v_{i,j} - v_{i-1,j}) + \sum_{i=0}^{n_1-1} \sum_{j=1}^{n_2-1} (u_{i,j} - u_{i,j-1})(v_{i,j} - v_{i,j-1}),$$

de sorte que l'on cherche à minimiser $\varphi_2(U,U)$ sous la contrainte que $U \in C$.

(b) Montrer que cela revient à minimiser $\frac{1}{2}\varphi_2(X,X) + \varphi_1(X)$, pour $X \in E$, où φ_1 est une forme linéaire $E \to \mathbb{R}$. Montrer que si $M \neq 0$ (au moins un pixel a été transmis), la forme φ_2 (bilinéaire symétrique) est définie positive sur E : $\varphi_2(X,X) > 0$ si $X \in E \setminus \{0\}$.

(c) On note $\langle \cdot, \cdot \rangle$ le produit scalaire canonique sur les matrices de $M_{n_1,n_2}(\mathbb{R})$, qui induit un produit scalaire sur E que l'on notera $\langle \cdot, \cdot \rangle_E$.

$$\langle G, H \rangle = \mathrm{Tr}(G^T H) = \sum_{i=0}^{n_1-1} \sum_{j=0}^{n_2-1} g_{i,j} h_{i,j}.$$

Montrer qu'il existe un endomorphisme $\mathcal{A}$ de E tel que pour tous X, Y dans E, on ait $\varphi_2(X,Y) = \langle X, \mathcal{A}(Y) \rangle_E$. Montrer qu'il existe $B \in E$ tel que $\varphi_1(X) = \langle B, X \rangle_E$ pour tout $X \in E$. En déduire qu'on a donc une solution unique au problème de minimisation (dès que $M \neq 0$).

Quelle serait la taille de la matrice pour stocker $\mathcal{A}$?

(d) Pourquoi cherche-t-on à pouvoir calculer $\mathcal{A}(X)$, sans stocker X sous la forme d'un vecteur (exprimé dans une base de E) et $\mathcal{A}$ sous la forme d'une matrice ? Montrer que pour toutes matrices X, Y de $M_{n_1,n_2}(\mathbb{R})$, on a

$$\varphi_2(X,Y) = \langle D_{n_1} X, D_{n_1} Y\rangle_{M_{n_1-1,n_2}} + \langle X D_{n_2}^T, Y D_{n_2}^T\rangle_{M_{n_1,n_2-1}},$$

où D_n est une matrice $(n-1)\times n$ nulle sauf sur la diagonale principale et la première diagonale supérieure. En déduire une expression de $\mathcal{A}(X)$ faisant intervenir les matrices X, $D_{n_1}^T D_{n_1}$, $D_{n_2}^T D_{n_2}$ et la multiplication élément par élément $\circledast$ avec $\overline{M}$.

Obtenir de la même manière une expression de B faisant intervenir les matrices F, $D_{n_1}^T D_{n_1}$, $D_{n_2}^T D_{n_2}$, l'opérateur $\circledast$, et $\overline{M}$.

(e) En déduire que l'on peut appliquer la méthode du gradient conjugué pour résoudre ce problème d'optimisation, et ce sans avoir à stocker une matrice A correspondant à l'application linéaire $\mathcal{A}$ écrite dans une base de E.

Pour améliorer la qualité de l'image rendue, on cherche à éviter certains problèmes qui sont liés au fait que la minimisation est quadratique (en particulier, les bords sont mal rendus). On modifie la fonction à optimiser, et on s'intéresse au problème suivant :

$$\inf_{U\in C} J(U), \qquad \text{avec } J(U) = \frac{1}{2}\sum_{i=1}^{n_1-1}\sum_{j=1}^{n_2-1}\sqrt{\delta + (u_{i,j}-u_{i-1,j-1})^2 + (u_{i,j-1}-u_{i-1,j})^2},$$

où δ est un paramètre de régularisation qu'on aimerait prendre assez petit

(f) En utilisant les mêmes méthodes que dans les questions précédentes, montrer que dans le premier cas on peut se ramener à un problème de minimisation sans contrainte par rapport à la variable $x \in E$.

Pourquoi ne peut-on plus appliquer la méthode de gradient conjugué pour ces deux problèmes ? Quelle méthode pourrait-on prendre ?

(g) * Calculer la dérivée partielle par rapport à $x_{i,j}$ de la fonction à optimiser. En déduire qu'on peut exprimer le gradient sans faire de multiplication matricielle.

3 Gradient conjugué pour les moindres carrés linéaires (examen de mai 2015)

Soient $M \in M_{m,n}(\mathbb{R})$ et $y \in \mathbb{R}^m$. Dans la suite $\|\cdot\|$ correspondra toujours à la norme euclidienne sur $\mathbb{R}^n$ ou $\mathbb{R}^m$, suivant le contexte. On s'intéresse au problème suivant :

$$\inf_{x\in\mathbb{R}^n} \|Mx - y\|. \tag{1}$$

(a) Montrer que le problème (1) admet toujours au moins une solution. Montrer que les solutions sont caractérisées par l'équation suivante (appelée « équation normale ») :

$$M^T M x = M^T y. \tag{2}$$

Indication : on pourra penser à une projection orthogonale sur un sous-espace vectoriel bien choisi.

On suppose pour la suite que la matrice M, vue comme une application de $\mathbb{R}^n$ dans $\mathbb{R}^m$, est injective.

(b) Pourquoi a-t-on nécessairement $m \geqslant n$? Montrer que la solution du problème (1) est unique.

(c) En écrivant $A = M^T M \in M_n(\mathbb{R})$ et $B = -M^T y \in \mathbb{R}^n$, montrer que le problème (1) revient à minimiser la fonction $f : x \mapsto \frac{1}{2}\langle x, Ax\rangle + \langle B, x\rangle$.
Cours : pourquoi peut-on appliquer la méthode du gradient conjugué, et dans quels cas a-t-on intérêt à le faire, plutôt que de résoudre directement le système linéaire (2) ?

(d) *Cours :* écrire l'initialisation et les équations standard donnant les itérées de la méthode du gradient conjugué pour cette fonction f, en fonction uniquement de M, y et d'un vecteur initial x_0. On notera comme dans le cours x_k pour le vecteur que l'on considère, r_k pour le gradient, p_k pour la direction de descente, et α_k le pas.

On s'intéresse maintenant à savoir quel est le bon ordre des opérations à faire pour calculer le vecteur $M^T M\, p_k$ dont on a besoin dans l'algorithme, pour un cas particulier de matrice creuse ne contenant que de l'ordre de n éléments non-nuls.

(e) * On considère pour $d_1,\ldots,d_n$ et $c_1,\ldots,c_n$ des réels non-nuls la matrice de $M_{n+1,n}(\mathbb{R})$ suivante :

$$M = \begin{pmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & d_n \\ c_1 & c_2 & \cdots & c_n \end{pmatrix}.$$

Montrer que M est injective et que $M^T M$ n'a que des éléments non-nuls. En déduire que pour calculer $(M^T M)\, p_k$, même en ayant déjà calculé $M^T M$, cela nécessite n^2 multiplications de réels. Montrer que pour calculer $M^T(M\, p_k)$ dans cet ordre, cela ne nécessite que $4n$ multiplications. Quelle est l'autre raison qui nous pousse à ne pas vouloir calculer initialement $M^T M$ et la stocker pour les calculs suivants, si n est grand ?

Si on a déjà calculé les vecteurs $M\, p_k$ et $M^T M\, p_k$, le calcul $\langle p_k, M^T M\, p_k\rangle = \|M\, p_k\|^2$ dont on a besoin pour le calcul de α_k nécessite n ou m opérations suivant la formule utilisée. On veut montrer qu'il y a un intérêt à utiliser la deuxième formule en terme de précision numérique, même si elle est plus coûteuse.

On rappelle que si η est la précision machine, alors l'erreur faite par la machine en calculant $a + b$ est d'ordre $\max(|a|, |b|)\eta$, et l'erreur faite en calculant ab est d'ordre $|ab|\eta$.

(f) * On prend $M = \begin{pmatrix} \varepsilon & 0 \\ 1 & 1 \end{pmatrix}$ et $p = \begin{pmatrix} 1 \\ -1+\delta \end{pmatrix}$ avec $0 < \varepsilon \ll 1$ et $0 < \delta \ll 1$. Si η est la précision machine et que $\eta \ll \varepsilon$ et $\eta \ll \delta$, alors montrer que l'erreur faite par la machine en calculant Mp, puis $\|Mp\|^2$ est d'ordre $O((\varepsilon^2 + \delta)\eta)$.

Montrer que l'erreur est bien plus grande, d'ordre $O(\eta)$, lorsque la machine calcule Mp, puis $M^T(Mp)$, puis $\langle p, M^T(Mp)\rangle$.